HAL open science

Algorithms for activity correction models for geochemical speciation and reactive transport modeling

Jérôme Carrayrou, Caroline Bertagnolli, Marwan Fahs

To cite this version:

Jérôme Carrayrou, Caroline Bertagnolli, Marwan Fahs. Algorithms for activity correction models for geochemical speciation and reactive transport modeling. AIChE Journal, 2022, 68 (1), pp.Bertagnolli. 10.1002/aic.17391 . hal-03526354

HAL Id: hal-03526354

https://hal.science/hal-03526354v1

Submitted on 14 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AIChE Journal

DOI10.1002/aic.17391

Algorithms for activity correction models for geochemical speciation and reactive transport modelling.

Jérôme Carrayrou[1,*], Caroline Bertagnolli[2], Marwan Fahs[1]

[1] *University of Strasbourg, Institut Terre et Environnement de Strasbourg; UMR 7063 UdS-CNRS*

[2] *University of Strasbourg, Reconnaissance et Procédés de Séparation Moléculaire, IPHC; UMR 7178 UdS-CNRS*

[*] *corresponding author: jerome.carrayrou@unistra.fr*

Accepted Article

Keywords

Geochemical modelling; instantaneous equilibrium chemistry; activity correction, outer fixed-point algorithm, Davis activity, Debye-Hückel activity, B-dot model.

Abstract

Reactive transport softwares are today one of the cornerstones of environmental research. They contain multiphysics with very complex algorithms, including flow, transport, chemical and sometimes heat transport, mechanical and/or biological algorithms. Because of this complexity, some parts of these algorithms still have not been sufficiently studied. In this work, we focus on algorithms for activity correction, a specific subset of equilibrium chemistry algorithms. We show that the most used algorithm (the inner fixed-point algorithm) or the most rigorous algorithm (the full Newton) might not be the most efficient, and we propose a new one, the outer fixed-point algorithm, which is more robust and faster than other algorithms.

Introduction

The problem of groundwater management has received increasing attention, and many tools have been developed to address this issue. One of these tools, reactive transport modeling, was first limited to laboratory experiments [1] and then extended to the comprehension of problems in various fields [2]. Reactive transport modeling is actually a mature research field that has produced important results in many environmental domains, such as water management, sea water intrusion [3], long-term nuclear waste storage [4], CO_2 sequestration [5, 6] and heavy metal contamination [7].

This article has been accepted for publication and undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process which may lead to differences between this version and the Version of Record. Please cite this article as doi: 10.1002/aic.17391© 2021 American Institute of Chemical Engineers
Received: Mar 06, 2021;Revised: Jun 23, 2021;Accepted: Jul 17, 2021

This article is protected by copyright. All rights reserved.

Accepted Article

Numerous reactive transport codes are available, and some review articles [6, 8–13] propose an overview of them.

General algorithms and/or numerical methods for geochemical speciation modelling are widely recognized as very challenging due to the non-linearity of the mathematic system, the very large range of magnitude of the chemical species concentrations and the equilibrium constants. Since first works [14–16], many authors worked on improving these algorithms and/or numerical methods [17–29]. An examination of these articles shows that all these codes include one or more activity correction models. Even though the different models of activity correction are usually well-detailed, the algorithmic method used to compute the activity coefficients and to incorporate these calculations into the entire chemistry algorithm is usually not given. We found mention of it only in the user's guide for CHESS [30] and PHREEQC [31] codes.

In this work, we will present the two algorithmic ways found in the literature to incorporate activity correction models into an entire geochemical speciation algorithm; propose a new one and compare these three algorithms. All these algorithms can be easily implemented into geochemical codes but they are not all equivalent and some leads to an increase of numerical instabilities. To discuss on activity correction algorithm, we need to briefly present an entire geochemical speciation algorithm. Although some approaches have abandoned formulation based on mass action law [32, 33] or resolution based on a "Newton" type method [29, 34, 35], we will apply our activity correction algorithms to this historical formulation, so as not to make it too cumbersome. Because we will focus on a specific part of the geochemical algorithm, we will not discuss, neither on the specific methods used to overpass the high non-linearity of the mathematical problem, nor on the ways to include specific chemical phenomena such as precipitation-dissolution, liquid-gas exchange or surface complexation modelling. This simplification made for the presentation of the model does not imply any loss of generality because the specificity neglected here are usually treated as an "external" loop in geochemical modelling [18, 28, 36, 37]. For example, precipitation-dissolution phenomena are usually solved as follow: (a) solving a first geochemical equilibrium with a given set of minerals. (b) Checking saturation of the existing and potential minerals. (c) Solving a new geochemical equilibrium including the existence of oversaturated (as calculated at step c) minerals and removing the under-saturated minerals. (d) Repeat (a), (b) and (c) to obtain a definitive set of minerals.

We first present the general concepts including fundamental equations (mass action law, activity correction model) and a simple formulation of the Newton method applied to geochemical speciation. The second part is devoted to the presentation of the two algorithms for activity correction found in literature and to the presentation of the new one. The specificities of these three

This article is protected by copyright. All rights reserved.

algorithms will be explained on a tricky test case. The third part of this article is devoted to benchmarking the studded algorithms against 3 different and challenging test cases. We can then conclude and propose some recommendations to improve the robustness and efficiency of geochemical speciation codes.

GENERAL CONCEPTS

Fundamental equations

A general formulation of a chemical reaction leading to the formation of one of the Nc species (C_i) from the number Nx of chosen components X_j depending on stoichiometric coefficients $a_{i,j}$ is written as:

$$C_i \rightleftarrows \sum_{j=1}^{Nx} a_{i,j} X_j \tag{1}$$

Components and species [14] may be named Primary and Secondary species [38] in some other works. We prefer the term "component" rather than "primary species". Saying "Primary species" implies that they are a subsets of the chemical species whereas the term "component" does not. By this way a component can be different from a chemical species. Some authors [39, 40] describe surface complexation phenomena by including the electrical potential at the surface as a component. In this work, we use the electron e^- as component to describe redox reaction in the iron-chromium test case.

Instantaneous equilibrium chemistry is usually described using two fundamental concepts: mass conservation equations and mass action laws. According to the classical formulation stated by Morel and Morgan [14], mass conservation equations describe the conservation of the total concentrations of the components (T_j), and mass action laws describe the formation of each chemical species as a combination of the Nx chosen components.

Mass conservation equations are written using the species concentrations [C_i] to conserve the total concentration T_j of each component:

$$T_j = \sum_{i=1}^{Nc} a_{i,j} \left[C_i\right] \tag{2}$$

On the other hand, mass action laws are written using the species {C_i}, the equilibrium constant K_i and components {X_j} of the activities:

Accepted Article

This article is protected by copyright. All rights reserved.

$$C_i = K_i \prod_{j=1}^{Nx} X_j^{a_{i,j}} \quad (3)$$

To ensure the closure of the system, an activity model is used. The activity coefficient (γ_i) allows determining the species activity from its concentration.

$$C_i = \gamma_i \left[C_i\right] \quad (4)$$

Several activity models have been developed, most of which use the ionic strength of the solution (I) and the electric charge of each species z_i:

$$I = \frac{1}{2} \sum_{i=1}^{Nc} z_i^2 \left[C_i\right] \quad (5)$$

One of the most used activity models for reactive transport is the Davies model, where the activity coefficients are given by:

$$\log \gamma_i = -A z_i^2 \left(\frac{\sqrt{I}}{1+\sqrt{I}} - BI \right) \quad (6)$$

The parameters A and B are defined by:

$$A = 1.82 \cdot 10^6 \ \varepsilon_W T^{-\frac{3}{2}} \text{ et } 0.2 \leq B \leq 0.3 \quad (7)$$

Because we found that the algorithmic behavior of the methods is quite independent on the activity model used, we only present here the Davies model and all the tests will be done using this one. Details on extended Debye-Hückel and B-dot model are given on appendix A-1.

Newton algorithm

The methodology for computing chemical equilibrium is well established [14, 15, 28, 30, 31, 41, 42] and is usually built on a Newton procedure. By incorporating the Nc mass action laws into the Nx conservation equation, one can obtain an Nx by Nx nonlinear system that must be iteratively solved. Here, we present the approach where mass action laws are written in a logarithmic form.

We define the logarithm of the components activity ξ_j as:

$$\xi_j = \ln X_j = \ln \gamma_j \left[X_j\right] \quad (8)$$

This can be rewritten as a matrix formulation:

$$\xi = diag \ \ln \gamma_X + \ln \mathbf{X} \quad (9)$$

Accepted Article

This article is protected by copyright. All rights reserved.

Where $diag\ \ln\gamma_X$ is a diagonal matrix, size Nx, Nx with $diag\ \ln\gamma_{X\ j,j} = \ln\ \gamma_j$. The mass action laws are then:

$$\ln\ C_i\ = \ln K_i + \sum_{j=1}^{Nx} a_{i,j}\xi_j \tag{10}$$

These can be rewritten by using the vector of species concentrations $\mathbf{C}$:

$$\ln\mathbf{C} = \ln\mathbf{K} - diag\ \ln\gamma_C\ + \mathbf{A}\times\xi \tag{11}$$

Where $diag\ \ln\gamma_C$ is a diagonal matrix, size Nc, Nc with $diag\ \ln\gamma_{C\ i,i} = \ln\ \gamma_i$. The conservation equations are:

$$\mathbf{T} = \mathbf{A}^T\mathbf{C} \tag{12}$$

Combining both equations, we obtain:

$$\mathbf{T} = \mathbf{A}^T \exp\ \ln\mathbf{K} - diag\ \ln\gamma_C\ + \mathbf{A}\xi \tag{13}$$

This nonlinear system is iteratively solved using a Newton procedure. At the nth iteration, the error is given by:

$$\mathbf{Y}^n = -\mathbf{T} + \mathbf{A}^T \exp\ \ln\mathbf{K} - diag\ \ln\gamma_C{}^n\ + \mathbf{A}\xi^n \tag{14}$$

By computing the derivative of the error versus the logarithm of the components of the activity, we obtain the Jacobian matrix $\mathbf{Z}$.

Historically, activity correction has been neglected when computing the Jacobian matrix. This approximation is justified if:

(i) The ionic strength is small enough so that the activity coefficients can be assumed to be equal to one (ideal solution).

(ii) The changes in ionic strength are sufficiently small during one Newton iteration so that the activity coefficients can be assumed to be constant.

The complete computation of the Jacobian matrix leads to equation (15):

$$\begin{aligned}
Z_{j,k} &= \frac{\partial Y_j}{\partial\xi_k} = \frac{\partial}{\partial\xi_k}\left(\sum_{i=1}^{Nc} a_{i,j}\cdot\exp\left(\ln K_i - \ln\gamma_i + \sum_{h=1}^{Nx} a_{i,h}\cdot\xi_h\right)\right)\\
Z_{j,k} &= \sum_{i=1}^{Nc} a_{i,j}\cdot\frac{\partial}{\partial\xi_k}\left(\exp\left(\ln K_i - \ln\gamma_i + \sum_{h=1}^{Nx} a_{i,h}\cdot\xi_h\right)\right)\\
Z_{j,k} &= \sum_{i=1}^{Nc} a_{i,j}\cdot\exp\left(\ln K_i - \ln\gamma_i + \sum_{h=1}^{Nx} a_{i,h}\cdot\xi_h\right)\cdot\frac{\partial}{\partial\xi_k}\left(\ln K_i - \ln\gamma_i + \sum_{h=1}^{Nx} a_{i,h}\cdot\xi_h\right)\\
Z_{j,k} &= \sum_{i=1}^{Nc} a_{i,j}\cdot\left[C_i\right]\cdot\frac{\partial}{\partial\xi_k}\left(\ln\gamma_i + a_{i,k}\cdot\xi_k\right)\\
Z_{j,k} &= \sum_{i=1}^{Nc} a_{i,j}\cdot\left[C_i\right]\cdot\left(a_{i,k} - \frac{\partial\ \ln\gamma_i}{\partial\xi_k}\right)
\end{aligned} \tag{15}$$

Accepted Article

This article is protected by copyright. All rights reserved.

This is because:

$$\frac{\partial}{\partial \xi_k}\left(\frac{K_i}{\gamma_i}\cdot \exp\left(\sum_{h=1}^{Nx} a_{i,h}\cdot \xi_h\right)\right) = \frac{K_i}{\gamma_i}\cdot a_{i,k}\cdot \exp\left(\sum_{h=1}^{Nx} a_{i,h}\cdot \xi_h\right) = a_{i,k}\cdot\left[c_i\right] \tag{16}$$

Neglecting the derivative of the activity coefficient (hypothesis (i) or (ii)), one obtains (17):

$$Z_{j,k} = \sum_{i=1}^{Nc} a_{i,j}\cdot a_{i,k}\cdot\left[c_i\right] \tag{17}$$

Alternatively, with $diag\ \mathbf{C}$ a diagonal matrix, size Nc, Nc and $diag\ \mathbf{C}_{i,i} = \left[c_i\right]$:

$$\mathbf{Z} = \mathbf{A}^T diag\ \mathbf{C}\ \mathbf{A} \tag{18}$$

Solving the linear system (19) gives the step of the Newton method $\Delta\xi^n$:

$$\mathbf{Z}^n \Delta\xi^n = -\mathbf{Y}^n \tag{19}$$

Then, a new value of the activity component is given by:

$$\xi^{n+1} = \xi^n + \Delta\xi^n \tag{20}$$

The procedure is repeated until the error is sufficiently small. In this work, we check the convergence using relation (21) with the convergence criteria $\varepsilon = 10^{-12}$. Relation (21) defines a relative error Err that ensures an equivalent accuracy for both major and trace element.

$$Err = \max\left(\frac{Y_j}{\left|T_j\right| + \sum_{i=1}^{Nc}\left|a_{i,j}\right| C_i}\right) \le \varepsilon \tag{21}$$

If convergence is not reached until a prescribed number of iterations, 1,000 in this work, the procedure is stopped and the method fails to converge. The robustness of a method reports its ability to reach the convergence within the maximum prescribed iterations.

Accepted Article

This article is protected by copyright. All rights reserved.

Accepted Article

POSSIBLE ALGORITHMS FOR ACTIVITY CORRECTION

We found two algorithms reported in the literature to handle activity correction: the inner fixed-point and the full Newton algorithm. We suppose this point is poorly reported for several reasons:

(i) Activity coefficients do not change a lot and are usually close to one. Because they depends on the ionic strength of the solution, they are supposed to be less modified during the optimization loops.

(ii) Other numerical difficulties are much more complicated to handle. Among them: the large range of magnitude of equilibrium constants, species and components concentrations; important nonlinearities of the system due to stoichiometric coefficients in the mass action laws; very high condition number of the Jacobian matrix.

(iii) Activity correction can be included in the global geochemical algorithm by a very simple way (see inner fixed-point algorithm).

Nevertheless, by working on numerical instabilities on computing geochemical equilibrium, we found that points (i) and (ii) are not verified for some specific chemical system. We then develop a third algorithm to fix these problems: the outer fixed-point algorithm.

Inner fixed-point activity algorithm

This algorithm is the most commonly used algorithm, according to the few information obtained. It is used by CHESS codes [30], SPECY [43] and we found it be examining the FORTRAN code of IMPACT [44].At each Newton loop, the activity coefficients are updated according to the new ionic strength (Figure 1). From a strictly mathematical point of view, this method is then a quasi-Newton method because the Jacobian matrix Z is an approximation of the derivative of the error using equation (17).

Full Newton activity algorithm

One can find an approach in the PHREEQC code [31] for including the derivative of the activity coefficients in the Jacobian matrix. This lead to the Full Newton algorithm (Figure *1*) where the Jacobian matrix is computed using equation (15).

The derivative of the activity coefficients must be calculated. Because the activity coefficients are strongly dependent on the ionic strength, we write:

$$\frac{\partial\left[\ln\ \gamma_i\right]}{\partial \xi_k} = \frac{\partial\left[\ln\ \gamma_i\right]}{\partial I}\frac{\partial I}{\partial \xi_k} \tag{22}$$

and then compute two parts:

(i) The derivative of the activity coefficients versus the ionic strength $\frac{\partial\left[\ln\ \gamma_i\right]}{\partial I}$. This component is model-dependent and simple to compute regardless of which model is used. Details for Davies model are given on equation (35) , details for extended Debye-Hückel and B-dot models are given on Appendix A-1.

This article is protected by copyright. All rights reserved.

(ii) The derivative of the ionic strength versus the logarithm of the activity components, which is independent on the activity correction model used.

We propose here two computation ways. A recursive formulation using equation (29) and an explicit formulation using equation (34). Both formulations lead to the same Jacobian matrix whatever the test case and only differs on the computing time. Except for CPU time comparison, we will use the explicit formulation.

Computation method 1: a recursive formulation

Computing the derivative of the ionic strength yields:

$$\frac{\partial I}{\partial \xi_k} = \frac{1}{2}\sum_{p=1}^{Nc} z_p^2 \cdot \frac{\partial\left[C_p\right]}{\partial \xi_k} = \frac{1}{2}\sum_{p=1}^{Nc} z_p^2 \cdot C_p \cdot \left(a_{p,k} - \frac{\partial\left[\ln\ \gamma_p\right]}{\partial \xi_k}\right) \tag{23}$$

We obtain a recursive formulation for the derivative of the activity coefficients:

$$\frac{\partial\left[\ln\ \gamma_i\right]}{\partial \xi_k} = \frac{\partial\left[\ln\ \gamma_i\right]}{\partial I} \times \frac{1}{2}\sum_{p=1}^{Nc} z_p^2 \cdot C_p \cdot \left(a_{p,k} - \frac{\partial\left[\ln\ \gamma_p\right]}{\partial \xi_k}\right) \tag{24}$$

This recursive formulation can be rearranged as:

$$\begin{aligned}\frac{\partial\left[\ln\ \gamma_i\right]}{\partial \xi_k} &= \frac{1}{2}\frac{\partial\left[\ln\ \gamma_i\right]}{\partial I} \times \sum_{p=1}^{Nc} z_p^2 \cdot a_{p,k} \cdot C_p - \frac{1}{2}\frac{\partial\left[\ln\ \gamma_i\right]}{\partial I} \times \sum_{p=1}^{Nc}\left(z_p^2 \cdot C_p \cdot \frac{\partial\left[\ln\ \gamma_p\right]}{\partial \xi_k}\right) \\ &= N_i + \sum_{p=1}^{Nc} M_{i,p} \cdot \frac{\partial\left[\ln\ \gamma_p\right]}{\partial \xi_k}\end{aligned} \tag{25}$$

The vector $\mathbf{N}_k$ contains the explicit portion of the equation. This vector differs depending on the component used to derive the activity coefficient through the stoichiometric coefficient $a_{p,k}$.

$$N_i = \frac{1}{2}\frac{\partial\left[\ln\ \gamma_i\right]}{\partial I} \times \sum_{p=1}^{Nc} z_p^2 \cdot a_{p,k} \cdot C_p \tag{26}$$

The matrix $\mathbf{M}$ contains the coefficients of the implicit portion of the equation. This matrix is the same regardless of the component used in the derivation.

$$M_{i,p} = -\frac{1}{2}\frac{\partial\left[\ln\ \gamma_i\right]}{\partial I} \times z_p^2 \cdot C_p \tag{27}$$

A complete derivative of the activity coefficients is given by the following linear system:

$$\frac{\partial}{\partial \xi_k}\left[\ln\ \gamma\right] = \mathbf{N}_k + \mathbf{M}\frac{\partial}{\partial \xi_k}\left[\ln\ \gamma\right] \text{ or } \qquad \mathbf{Id} - \mathbf{M}\ \frac{\partial}{\partial \xi_k}\left[\ln\ \gamma\right] = \mathbf{N}_k \tag{28}$$

Accepted Article

This article is protected by copyright. All rights reserved.

$$\frac{\partial}{\partial \xi_k}\left[\ln\ \gamma\right] = \mathbf{Id} - \mathbf{M}^{-1}\mathbf{N}_k \tag{29}$$

Computation method 2: an explicit formulation

Computing the derivative of the ionic strength yields:

$$\frac{\partial I}{\partial \xi_k} = \frac{1}{2}\sum_{p=1}^{Nc} z_p^2 \cdot \frac{\partial\left[C_p\right]}{\partial \xi_k} = \frac{1}{2}\sum_{p=1}^{Nc} z_p^2 \cdot C_p \cdot \left(a_{p,k} - \frac{\partial\left[\ln\ \gamma_p\right]}{\partial \xi_k}\right) \tag{30}$$

$$\frac{\partial I}{\partial \xi_k} = \frac{1}{2}\sum_{p=1}^{Nc} z_p^2 \cdot C_p \cdot \left(a_{p,k} - \frac{\partial\left[\ln\ \gamma_p\right]}{\partial I}\frac{\partial I}{\partial \xi_k}\right) = \frac{1}{2}\sum_{p=1}^{Nc} z_p^2 \cdot a_{p,k} \cdot C_p - \frac{1}{2}\frac{\partial I}{\partial \xi_k} \cdot \sum_{p=1}^{Nc} z_p^2 \cdot C_p \cdot \frac{\partial\left[\ln\ \gamma_p\right]}{\partial I} \tag{31}$$

$$\frac{\partial I}{\partial \xi_k}\left(1 + \frac{1}{2} \cdot \sum_{p=1}^{Nc} z_p^2 \cdot C_p \cdot \frac{\partial\left[\ln\ \gamma_p\right]}{\partial I}\right) = \frac{1}{2}\sum_{p=1}^{Nc} z_p^2 \cdot a_{p,k} \cdot C_p \tag{32}$$

We obtain an explicit formulation for the derivative of the ionic strength:

$$\frac{\partial I}{\partial \xi_k} = \frac{\sum_{p=1}^{Nc} z_p^2 \cdot a_{p,k} \cdot C_p}{2 + \sum_{p=1}^{Nc} z_p^2 \cdot C_p \cdot \frac{\partial\left[\ln\ \gamma_p\right]}{\partial I}} \tag{33}$$

This allows for an explicit formulation of the derivative of the activity coefficients:

$$\frac{\partial\left[\ln\ \gamma_i\right]}{\partial \xi_k} = \frac{\partial\left[\ln\ \gamma_i\right]}{\partial I}\frac{\partial I}{\partial \xi_k} = \frac{\partial\left[\ln\ \gamma_i\right]}{\partial I}\frac{\sum_{p=1}^{Nc} z_p^2 \cdot a_{p,k} \cdot C_p}{2 + \sum_{p=1}^{Nc} z_p^2 \cdot C_p \cdot \frac{\partial\left[\ln\ \gamma_p\right]}{\partial I}} \tag{34}$$

Derivative of the activity coefficients versus the ionic strength $\frac{\partial\left[\ln\ \gamma_i\right]}{\partial I}$: Davies model

For a Davies model, the activity coefficients are given in logarithmic form by:

$$\ln\ \gamma_i = -\ln\ 10\ Az_i^2\left(\frac{\sqrt{I}}{1+\sqrt{I}} - BI\right) \tag{6}$$

Taking the derivative of this equation yields:

$$\frac{\partial\left[\ln\ \gamma_i\right]}{\partial I} = -\ln\ 10\ Az_i^2\frac{\partial}{\partial I}\left(\frac{\sqrt{I}}{1+\sqrt{I}} - BI\right) = -\ln\ 10\ Az_i^2\left[\frac{1}{2\sqrt{I}}\frac{1}{1+\sqrt{I}^{\,2}} - B\right] \tag{35}$$

Accepted Article

This article is protected by copyright. All rights reserved.

Accepted Article

Derivative of the activity coefficients versus the ionic strength for the extended Debye-Hückel and the B-dot models can be found on Appendix A-1.

Outer fixed-point activity algorithm

Here, we propose this new algorithm. For a given set of values for the activity coefficients, the Newton procedure is iteratively run until convergence is achieved (Loop 1 on Figure 1). Once convergence is reached, the activity coefficients are updated according to the ionic strength computed at the time of convergence (Loop 2 on Figure 1). This procedure is repeated until no changes are detected neither on the component activities (Test convergence = YES on Figure 1) nor on the activity coefficients (Activity changed = YES on Figure 1). We then obtain a true Newton algorithm where the Jacobian matrix calculated by equation (17) is truly the derivative of the error.

Comparison of the algorithms on a trivial test case

To compare the activity correction algorithms without any interferences of the other numerical difficulties reported to the Newton procedure, we propose a test case without any chemical reactions (Table *1*) called "test case with only activity correction" (details can be found appendix A-2). Any nonlinearity is only due to activity correction. The chemical system is composed of chloride Cl^- ions, calcium Ca^{2+} ions, aluminum Al^{3+} ions and tin Sn^{4+} ions. We neglect water dissociation and all chemical reactions. The details and equilibrium solutions are given in appendix A-2.

One should note that this test case is not chemically realistic. Moreover, its numerical value comes only from activity correction if the unknowns of the nonlinear system (14) are the logarithms of the activity components ξ_j (8). Otherwise, if the unknowns are component concentrations $\left[X_j\right]$, the problem becomes trivial and linear, and its solution is the total concentration $\left[X_j\right] = T_j \quad \forall j \in 1,\ldots,Nx$.

We present two scenarios for the test case with *only activity correction*: one with a low ionic strength and one with a high ionic strength. The objective is to determine the influence of the activity correction on the Newton procedure depending on the algorithm used. For the situation with a low ionic strength, this influence is expected to be negligible, whereas we expect a greater impact in the situation with a high ionic strength.

For the low ionic strength situation, the initial component activities are 5.0 10^{-7} mol l^{-1} for all components. The ionic strength is 7.80 10^{-6} mol $l^{-1,}$ and we obtain the species concentrations and activity values, which are given in Table *2*. Also in Table *2*, we show the first Newton steps $\Delta\xi$ proposed by the fixed-point algorithms (inner and outer) and by the full Newton algorithm.

This article is protected by copyright. All rights reserved.

The Jacobian matrices given by both fixed-point algorithms are the same:

$$Z = \begin{bmatrix} 5.00 \cdot 10^{-7} & 0.00 & 0.00 & 0.00 \\ 0.00 & 5.00 \cdot 10^{-7} & 0.00 & 0.00 \\ 0.00 & 0.00 & 5.00 \cdot 10^{-7} & 0.00 \\ 0.00 & 0.00 & 0.00 & 5.00 \cdot 10^{-7} \end{bmatrix} \tag{36}$$

The full Newton algorithm leads to the following derivative matrix of the activity coefficients:

$$\frac{d \ \ln \gamma_i}{d\xi_k} = \begin{bmatrix} -5.46 \cdot 10^{-5} & -2.21 \cdot 10^{-4} & -5.05 \cdot 10^{-4} & -9.19 \cdot 10^{-4} \\ -2.18 \cdot 10^{-4} & -8.83 \cdot 10^{-4} & -2.02 \cdot 10^{-3} & -3.67 \cdot 10^{-3} \\ -4.91 \cdot 10^{-4} & -1.99 \cdot 10^{-3} & -4.54 \cdot 10^{-3} & -8.27 \cdot 10^{-3} \\ -8.74 \cdot 10^{-4} & -3.53 \cdot 10^{-3} & -8.08 \cdot 10^{-3} & -1.47 \cdot 10^{-2} \end{bmatrix} \tag{37}$$

The following Jacobian matrix is also obtained:

$$Z = \begin{bmatrix} 5.02 \cdot 10^{-7} & 1.11 \cdot 10^{-10} & 2.53 \cdot 10^{-10} & 4.61 \cdot 10^{-10} \\ 1.11 \cdot 10^{-10} & 5.07 \cdot 10^{-7} & 1.02 \cdot 10^{-9} & 1.86 \cdot 10^{-9} \\ 2.53 \cdot 10^{-10} & 1.02 \cdot 10^{-9} & 5.18 \cdot 10^{-7} & 4.26 \cdot 10^{-9} \\ 4.61 \cdot 10^{-10} & 1.86 \cdot 10^{-9} & 4.26 \cdot 10^{-9} & 5.36 \cdot 10^{-7} \end{bmatrix} \tag{38}$$

For the scenario with a low ionic strength, the Jacobian matrices are quite equivalent for the full Newton and fixed-point algorithms, leading to similar Newton steps ($\Delta\xi$).

The high ionic strength scenario starts with an initial component activities equal to 5.0 10^{-4} mol.l^{-1} for all components. The ionic strength equals 1.197 mol.l^{-1}. The species concentrations, activity values and first Newton steps $\Delta\xi$ proposed by the fixed-point algorithm (inner and outer) and full Newton algorithm are given in Table *3*.

The Jacobian matrices given by both fixed point algorithms are the same:

$$Z = \begin{bmatrix} 5.00 \cdot 10^{-4} & 0.00 & 0.00 & 0.00 \\ 0.00 & 5.00 \cdot 10^{-4} & 0.00 & 0.00 \\ 0.00 & 0.00 & 5.00 \cdot 10^{-4} & 0.00 \\ 0.00 & 0.00 & 0.00 & 5.00 \cdot 10^{-4} \end{bmatrix} \tag{39}$$

The full Newton algorithm leads to the following derivative matrix of the activity coefficients:

Accepted Article

This article is protected by copyright. All rights reserved.

$$\frac{d\ \ln\gamma_i}{d\xi_k} = \begin{bmatrix} 1.43\cdot10^{-5} & 1.65\cdot10^{-4} & 2.17\cdot10^{-3} & 4.58\cdot10^{-2} \\ 5.73\cdot10^{-5} & 6.61\cdot10^{-4} & 8.70\cdot10^{-3} & 1.83\cdot10^{-1} \\ 1.29\cdot10^{-4} & 1.49\cdot10^{-3} & 1.96\cdot10^{-2} & 4.12\cdot10^{-1} \\ 2.29\cdot10^{-4} & 2.64\cdot10^{-3} & 3.48\cdot10^{-2} & 7.33\cdot10^{-1} \end{bmatrix} \tag{40}$$

The following Jacobian matrix is also obtained:

$$Z = \begin{bmatrix} 7.12\cdot10^{-4} & -1.18\cdot10^{-7} & -1.55\cdot10^{-6} & -3.26\cdot10^{-5} \\ -1.18\cdot10^{-7} & 2.05\cdot10^{-3} & -1.79\cdot10^{-5} & -3.76\cdot10^{-4} \\ -1.55\cdot10^{-6} & -1.79\cdot10^{-5} & 1.18\cdot10^{-2} & -4.95\cdot10^{-3} \\ -3.26\cdot10^{-5} & -3.76\cdot10^{-4} & -4.95\cdot10^{-3} & 3.80\cdot10^{-2} \end{bmatrix} \tag{41}$$

For the scenario with a high ionic strength, the Jacobian matrices Z are very different for the full Newton and fixed-point algorithms, leading to very different Newton steps ($\Delta\xi$). Moreover, we find an increase in the condition number of matrix Z for the full Newton algorithm. The condition number of the matrix (41) is 21.6, whereas it equals 1 for matrix (39).

The symmetry of the Jacobian matrix in equations (38) and (41) is specific to this test case. As shown in equation (18), the Z matrix for the fixed-point algorithms is symmetric, whereas equation (15) proves that it is usually not symmetric for the full Newton algorithm.

We solved this test case with only activity correction using the three presented algorithms with the low and high ionic strength scenario. We plot on a graph (Figure 2) the evolution of the norm of the error (eq 16) versus the Newton iterations.

Comparing the three algorithms on the test case with *only activity correction*, one can see in Figure 2 that:

- The outer fixed-point algorithm requires the fewest Newton iterations to reach convergence, whereas the inner fixed-point algorithm requires the most iterations. The full Newton algorithm requires an intermediate number of Newton iterations.
- For the three algorithms, obtaining the solution at a low ionic strength requires less Newton iteration than at a high ionic strength. This point is obvious: for this test case, activity corrections are the only nonlinearity of the problem, and they are less important at a low ionic strength than at a high ionic strength.
- The outer fixed-point algorithm runs 3 minimization loops (Loop 2 on Figure 1) for situations with both low and high ionic strength. The first loops converge at 10 (low) and 23 (high) iterations; the second loop converges at 15 (low) and 29 (high) iterations. The third loop is the *confirmation* loop used to check that no changes in the ionic strength computation occur and then to confirm the convergence of the algorithm (Test convergence is YES and activity changed is YES on Figure *1*)

Accepted Article

This article is protected by copyright. All rights reserved.

It seems that the outer fixed point algorithms converges at lower value than others but it's an illusion. Convergence criteria ε is set to 10^{-12} for the three algorithms (eq 19). This illusion is because the convergence rate of the outer fixed point algorithms is higher than other algorithms near the solution, leading to a sharper slope of it's curve on Figure 2).

Accepted Article

This article is protected by copyright. All rights reserved.

Accepted Article

BENCHMARKING OF THE THREE ALGORITHMS

Testing procedure

It is well-known that the initial guess of the values of the components plays a critical role in the convergence of the Newton methods [13, 21]. To test several initial guesses, we generate a large number (30 000) of activity component values according to the following procedure:

$$\log X = 1 - rand \times \log X_{\min} + rand \times \log X_{\max} \quad (42)$$

where $rand \in [0;1]$ and $X_{\min}$ and $X_{\max}$ are given in the description of the chemical test case to handle a representative range of concentrations.

We then obtain 30 000 realizations of the optimization procedure using the same chemical test case but different initial guesses. To analyze this large amount of data, we construct a frequency graph of the number of Newton iterations needed to reach convergence. We plot graphs of the cumulative ratio of the resolutions that converge within a given number of Newton iterations. It gives the ratio of simulations that converge within a given number of Newton iteration. According to the graph, the algorithm that reaches a cumulative frequency of 1 is said to be robust. The algorithm that reaches a high cumulative frequency for a low number of Newton iterations is said to be fast. The difference between 1 and the highest cumulative frequency gives the failure ratio of the algorithm.

Chemical test cases

We propose 5 chemical test cases including the only activity correction test case presented previously. This selection includes a quite easy test (phosphoric acid), a quite difficult one (gallic acid), one very hard (iron-chromium) and one of very large size (Morel-Morgan).

Phosphoric acid test case

This test presents reactions between phosphoric acid and salt water. It includes 4 components and 8 chemical species. We handle only acid-base reactions: water dissociation and the 3 phosphoric acid reactions. This test case is numerically a simple chemical test, which is interesting in this context because of the PO_4^{3-} species which leads to important activity correction due to its -3 electrical charge. A table including the stoichiometric coefficients, equilibrium constants, total concentrations and equilibrium solutions is given in appendix A-3.

Gallic acid test case

This test case was proposed by Brassard and Bodurtha [36]. It includes 3 components and 17 chemical species. It is a classical test case, and many difficulties in convergence have been reported while solving it by using Newton or Newton-like algorithms [28, 29, 41, 45]. It should be selective according to the activity correction algorithm because of the presence of many highly charged

This article is protected by copyright. All rights reserved.

species such as AlL_3^{6-}. A table including the stoichiometric coefficients, equilibrium constants, total concentrations and equilibrium solutions is given in appendix A-4.

Iron-chromium test case

This test case concerns the rehabilitation of chromium-contaminated industrial soil using an iron-chromium reduction [2, 46]. Chromium (VI), which is the most toxic and mobile form of chromium, is reduced by iron (II) to yield chromium (III), which has a much lower solubility and is less toxic [47]. This test is reported to be a very difficult one [41, 45], so here we use some favorable testing conditions to increase the convergence of the Newton algorithm. Its numerical difficulties are attributed to the redox reactions described that lead to a large range of stoichiometric coefficients (from -4 to +8) and equilibrium constants (from $10^{-83.17}$ to $10^{80.90}$). Highly charged species such as Cr^{3+} or $Fe_2(OH)_2^{4+}$ will be challenging for activity coefficients algorithms. It includes 7 components and 39 species. A table including the stoichiometric coefficients, equilibrium constants, total concentrations and equilibrium solutions is given in appendix A-5.

Morel-Morgan test case

This test has been presented in the article that founded Newton algorithm as a key tool of equilibrium computation [14]. It is one of the larger test case reported in the literature, using 52 components to generate 781 species. It includes highly charged species such as $P_3O_{10}^{5-}$, a large range of stoichiometric coefficients (from -4 to +6) and equilibrium constants (from $10^{-31.7}$ to 10^{66}). A table including the stoichiometric coefficients, equilibrium constants, total concentrations and equilibrium solutions is given in appendix A-6.

Frequency graphs

Test case with *only activity correction*

The test case with *only activity correction* is a simple chemical test case. It makes sense only for studying the activity correction algorithms. It is solved by all the algorithms (see Table 5) within 150 Newton iterations (Figure 3). The fastest algorithm is the outer fixed-point algorithm, regardless of the ionic strength. Moreover, this algorithm shows a very low sensitivity to the ionic strength by resolving the low ionic strength case within 24 or 25 iterations and the high ionic strength case within 21 iterations regardless of the initial guess. The inner fixed-point and the full Newton algorithms are much more sensitive to the ionic strength, with significant increases in the number of iterations required to converge in the case with a high ionic strength. For this case, we find that the best algorithm is the outer fixed-point algorithm, and the inner fixed-point algorithm is the worst according to the number of Newton iterations. Taking the computing time of one Newton iteration into account (Table 4), we see that the full Newton algorithm is the slowest and the outer fixed-point algorithm is the fastest.

Accepted Article

This article is protected by copyright. All rights reserved.

Accepted Article

Phosphoric acid test case

. As said, this test case is quite easy to solve. We find that all the algorithms reach 100 % resolution (Table 5) in fewer than 70 iterations. Nevertheless, the presence of highly charged species such as PO_4^{3-} allows to differentiate the behavior of the algorithms. The frequency graph shows that the full Newton algorithm usually converges within the fewest iterations and that the inner fixed-point algorithm requires the most iterations to converge. Nevertheless, the outer fixed-point algorithm provides very interesting results for this test case: it converges within 36 and 39 iterations regardless of the initial guess.

Gallic acid test case

The results for the gallic acid test case (Figure 5) confirm those of the phosphoric acid test case. The inner fixed-point algorithm requires the most iterations to converge (approximately 150). The full Newton algorithm usually requires fewer iterations (approximately 130) but sometimes requires many more iterations (200); sometimes it fails to converge at all (0.05 % failure rate, see Table 5). The outer fixed-point algorithm converges with the fewest iterations (77 iterations maximum). Moreover, this algorithm gives the sharpest frequency graph, indicating that it is not sensitive to the initial guess.

Iron-chromium test case

The iron-chromium test case is the strongest test used here. It is reported [41, 45] to generate very high condition numbers. The range of initial guesses is chosen with the goal of keeping this test reasonable. Nevertheless, one can see (Figure 6) that it is very hard for the full Newton algorithm to converge, with a 91.35 % failure rate (Table 5). Notably, this full Newton algorithm sometimes results in faster resolutions (according to the number of Newton iterations) because it is the only algorithm that sometimes converges with fewer than 200 Newton iterations. Both the inner and outer fixed-point algorithms converge most frequently within 230-240 iterations. The inner fixed-point algorithm reaches 80 % of its realizations after 240 iterations but needs up to 1000 iterations to complete the set and fails to converge at a rate of 1.62 % (Table 5). The outer fixed-point algorithm requires between 226 and 233 iterations to converge regardless of the initial point. Moreover, it always succeeds in solving this test case.

Morel-Morgan test case

The Morel-Mogran test case is the largest test used here. It can be seen on Figure 7 that the inner fixed-point algorithm solves quite fast 40 % of the simulations using less than 200 iterations each. But this algorithm needs more and more iterations to solve 40 % of the other simulation and fails to converge for 17.91 % of the cases (Table 5). The Full Newton algorithm produces more regular results succeeding to solve easiest simulation within less than 100 iteration and using all the 1000 iterations to solve some of the hardest simulations. It fails to converge at the rate of 3.87 % (Table 5). The Outer fixed point algorithm is very efficient for this test case too, solving all the cases within less than 300 Newton iteration each.

This article is protected by copyright. All rights reserved.

Accepted Article

Comparison of CPU times and failure ratios

As expected, the CPU times for one iteration loop increase as the number of species increases in the test case and as the complexity of the algorithm increases (see Table 4). We show that the outer fixed-point algorithm is the fastest and that the full Newton algorithm is the slowest. Nevertheless, CPU times for the outer and the inner fixed-point algorithms are very close and their differences can be considered as negligible. Moreover, the recursive formulation of the full Newton algorithm is the slowest. For this reason, we strongly recommend not using the recursive formulation, and we prefer the explicit formulation.

We find that the only algorithm that solves all the test cases with a 100 % success rate is the outer fixed-point algorithm. The inner fixed-point algorithm is less robust and fails to solve the iron-chromium test 1.62 % of the time and the Morel-Morgan one 17.91 % of the time. The weakest algorithm is the full Newton algorithm, which fails at a 0.05 % rate for the gallic acid test, at a 3.87 % rate for the Morel-Morgan test and at a 91.35 % rate for the iron-chromium test.

CONCLUSION

Algorithms to handle activity correction into geochemical models have been poorly mentioned in the literature. In this work, we extensively described the 2 algorithms we found mentioned and presented a new one. We compared 3 algorithms based on their ability to handle activity correction in equilibrium chemistry solvers. They were compared on 5 chemical test cases selected according to their increasing numerical difficulty and/or size

The full Newton algorithm is the most integrated algorithm from a mathematical point of view. Nevertheless, we found it to be the slowest and weakest algorithm. We suppose this algorithm increases the nonlinearity of the chemical system by injecting activity corrections into the mass action equations and conservation laws. It increases the condition number of the Jacobian matrix, as shown by comparing (39) and (41). It has been shown [41, 45] that a condition number that is too high leads to inaccurate steps $\Delta\xi$ (19) in the Newton methods, leading to numerical difficulties or nonconvergence. Because chemical equilibrium computation is still a highly nonlinear problem, increasing its nonlinearity by injecting activity correction seems to be an inefficient choice.

The inner fixed-point algorithm includes an intermediate integration of activity correction into the Newton loop. Both loops, Newton for the mass action equations and conservation laws and fixed-point for activity correction, run together. In this way, changes induced by activity correction disturb the Newton minimization. This point explains the convergence difficulties of the inner fixed-point algorithm when activity correction becomes important.

The outer fixed-point algorithm proposes a complete separation between the Newton and activity correction loops. By this way, nonlinearity induced by activity correction cannot disturb the Newton convergence, and the condition number of the Jacobian matrix is lower than that obtained by the full Newton algorithm. This leads to a more stable and robust algorithm. We found that the outer fixed-

This article is protected by copyright. All rights reserved.

point algorithm is the fastest in terms of CPU times for one Newton iteration, usually faster than or equivalent to the other algorithms in terms of the number of required Newton iterations and the most robust.

We have thus shown that the external fixed point algorithm provides greater robustness and speed than the other two. The influence of the choice of the activity correction algorithm is amplified in this work, because the general geochemical equilibrium resolution algorithm used here is extremely simple. The use of specific numerical methods to reduce or overcome the strong non-linearities of geochemical systems, as implemented in modern codes, will obviously make this influence less visible. Nevertheless, and according to the results presented here, we recommend the outer fixed-point algorithm. This algorithm is the least time consuming for one Newton iteration, it usually requires the fewest number of iterations, and it is the most robust and least sensitive to the initial guess. Moreover, its implementation with existing geochemical algorithms is very simple and requires very few modifications.

Accepted Article

This article is protected by copyright. All rights reserved.

Accepted Article

Acknowledgements

The authors acknowledge the French programme NEEDS for its financial support to the project NewSolChem. We greatly thanks the anonymous reviewers and Michel Kern for their helpful comments.

Bibliography

1. Lefèvre F, Sardin M, Schweich D (1993) Migration of strontium in clayey and calcareous sandy soil: Precipitation and ion exchange. J Contam Hydrol 13:215–229. https://doi.org/10.1016/0169-7722(93)90058-Z

2. Wanner C, Eggenberger U, Mäder U (2012) A chromate-contaminated site in southern Switzerland – Part 2: Reactive transport modeling to optimize remediation options. Appl Geochem 27:655–662. http://dx.doi.org/10.1016/j.apgeochem.2011.11.008

3. Werner AD, Bakker M, Post VEA, et al (2013) Seawater intrusion processes, investigation and management: Recent advances and future challenges. Adv Water Resour 51:3–26. http://dx.doi.org/10.1016/j.advwatres.2012.03.004

4. De Windt L, Burnol A, Montarnal P, van der Lee J (2003) Intercomparison of reactive transport models applied to UO2 oxidative dissolution and uranium migration. J Contam Hydrol 61:303–312

5. Navarre-Sitchler AK, Maxwell RM, Siirila ER, et al (2013) Elucidating geochemical response of shallow heterogeneous aquifers to CO2 leakage using high-performance computing: Implications for monitoring of CO2 sequestration. Adv Water Resour 53:45–55. http://dx.doi.org/10.1016/j.advwatres.2012.10.005

6. Pruess K, Garcia J, Kovscek T, et al (2004) Code intercomparison builds confidence in numerical simulation models for geologic disposal of CO2. Energy 29:1431–1444

7. Wanner C, Druhan JL, Amos RT, et al (2015) Benchmarking the simulation of Cr isotope fractionation. Comput Geosci 19:497–521. https://doi.org/10.1007/s10596-014-9436-9

8. Steefel CI, Appelo CAJ, Arora B, et al (2015) Reactive transport codes for subsurface environmental simulation. Comput Geosci 19:445–478. https://doi.org/10.1007/s10596-014-9443-x

9. Yeh GT, Tripathi VS (1989) A critical evaluation of recent developments in hydrogeochemical transport models of reactive multichemical components. Water ResourRes **25**:93–108

10. Yaron B, Dror I, Berkowitz B (2010) Contaminant geochemistry-a new perspective. Naturwissenschaften 97:1–17

11. Steefel CI (2000) New directions in hydrogeochemical transport modeling: Incorporating multiple kinetic and equilibrium reaction pathways. Comput Methods Water Resour 1:331–338

12. van der Lee J, De Windt L (2001) Present state and future directions of modeling of geochemistry in hydrogeological systems. J Contam Hydrol 47:265–282

This article is protected by copyright. All rights reserved.

Accepted Article

13. Steefel CI, DePaolo DJ, Lichtner PC (2005) Reactive transport modeling: An essential tool and a new research approach for the Earth sciences. Earth Planet Sci Lett 240:539–558

14. Morel F, Morgan J (1972) Numerical method for computing equilibriums in aqueous chemical systems. Environ Sci Technol 6:58–67. https://doi.org/10.1021/es60060a006

15. Westall JC, Zachary JL, Morel FMM (1976) MINEQL: a computer program for the calculation of chemical equilibrium composition of aqueous system. Cambridge

16. Wigley TML (1977) WATSPEC: A computer program for determining the equilibrium speciation of aqueous solutions.

17. Zhang F, Yeh G-T (George), C. Parker J (2012) Reactive Transport Modeling in Variably Saturated Media with MIN3P: Basic Model Formulation and Model Enhancements. In: Groundwater Reactive Transport Models. BENTHAM SCIENCE PUBLISHERS, pp 186–211

18. Steefel C (2016) CRUNCHFLOW: Software for Modeling Multicomponent Reactive Flow and Transport. User's manual. USA: Lawrence Berkeley Laboratory

19. Steefel CI, Appelo CAJ, Arora B, et al (2015) Reactive transport codes for subsurface environmental simulation. Comput Geosci 19:445–478. https://doi.org/10.1007/s10596-014-9443-x

20. Parkhurst DL, Wissmeier L (2015) PhreeqcRM: A reaction module for transport simulators based on the geochemical model PHREEQC. Adv Water Resour 83:176–189. http://dx.doi.org/10.1016/j.advwatres.2015.06.001

21. Hoffmann J, Krautle S, Knabner P (2012) A general reduction scheme for reactive transport in porous media. Comput Geosci 16:1081–1099. https://doi.org/10.1007/s10596-012-9304-4

22. Poulet T, Gross L, Georgiev D, Cleverley J (2012) escript-RT: Reactive transport simulation in Python using escript. Comput Geosci 45:168–176. http://dx.doi.org/10.1016/j.cageo.2011.11.005

23. Xu T, Spycher N, Sonnenthal E, et al (2011) TOUGHREACT Version 2.0: A simulator for subsurface reactive transport under non-isothermal multiphase flow conditions. Comput Geosci 37:763–774. http://dx.doi.org/10.1016/j.cageo.2010.10.007

24. Lichtner PC (2007) FLOTRAN users manual: two-phase nonsothermal coupled thermal-hydrologic-chemical (THC) reactive flow and transport code version 2. Los Alamos National Laboratory, New Mexico, USA

25. van der Lee J, De Windt L, Lagneau V, Goblet P (2003) Module-oriented modeling of reactive transport with HYTEC. Comput Geosci 29:265–275

26. Meeussen JCL (2003) ORCHESTRA: An Object-Oriented Framework for Implementing Chemical Equilibrium Models. Environ Sci Technol 37:1175–1182. https://doi.org/10.1021/es025597s

27. Paz-García JM, Johannesson B, Ottosen LM, et al (2013) Computing multi-species chemical equilibrium with an algorithm based on the reaction extents. Comput Chem Eng 58:135–143. http://dx.doi.org/10.1016/j.compchemeng.2013.06.013

28. Carrayrou J, Mosé R, Behra P (2002) New efficient algorithm for solving thermodynamic chemistry. AIChE J 48:894–904

This article is protected by copyright. All rights reserved.

Accepted Article

29. Nazer SA, Jazar M, Rosier C (2021) Convergence acceleration of iterative sequences for equilibrium chemistry computations. Comput Geosci. https://doi.org/10.1007/s10596-021-10054-1

30. Van der Lee J (1998) Thermodynamic and mathematical concept of CHESS. Ecole des Mines de Paris

31. Parkhurst DL, Appelo CAJ (1999) User's guide to PHREEQC (version 2)- A computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations. *Denver, CO, USA.*

32. Leal AMM, Blunt MJ, LaForce TC (2014) Efficient chemical equilibrium calculations for geochemical speciation and reactive transport modelling. Geochim Cosmochim Acta 131:301–322. https://doi.org/10.1016/j.gca.2014.01.038

33. Allan M. M. Leal, Dmitrii A. Kulik, William R. Smith and Martin O. Saar (2017) An overview of computational methods for chemical equilibrium and kinetic calculations for geochemical and reactive transport modeling. Pure Appl Chem 89:597–643. https://doi.org/10.1515/pac-2016-1107

34. Amir L, Kern M (2021) Jacobian Free Methods for Coupling Transport with Chemistry in Heterogenous Porous Media. Water 13:370. https://doi.org/10.3390/w13030370

35. Al Nazer S (2020) Sur la modélisation du transport réactif dans les réserves d'eaux potables. Université du Littoral Côte d'Opale ; Ecole Doctorale des Sciences et de Technologie (Beyrouth)

36. Brassard P, Bodurtha P (2000) A feasible set for chemical speciation problems. Comput Geosci 26:277–291

37. van der Lee J, De Windt L, Lagneau V, Goblet P (2003) Module-oriented modeling of reactive transport with HYTEC. React Transp Model Geosci 29:265–275. https://doi.org/10.1016/S0098-3004(03)00004-9

38. Lichtner PC (1995) Principles and practice of reactive transport modeling. pp 117–130

39. Lützenkirchen J (1999) The constant capacitance model and variable ionic strength: An evaluation of possible applications and applicability. J Colloid Interface Sci 217:8–18

40. Lützenkirchen J (1998) Parameter estimation for the triple layer model. Analysis of conventional methods and suggestion of alternative possibilities. J Colloid Interface Sci 204:119–127

41. Marinoni M, Carrayrou J, Lucas Y, Ackerer P (2017) Thermodynamic equilibrium solutions through a modified Newton Raphson method. AIChE J 63:. https://doi.org/10.1002/aic.15506

42. Steefel C (2006) Crunch – user's guide. USA: Lawrence Berkeley Laboratory

43. Carrayrou J (2010) Looking for some reference solutions for the reactive transport benchmark of MoMaS with SPECY. Comput Geosci 14:393–403

44. Jauzein M, Andre C, Margrita R, et al (1989) A flexible computer code for modelling transport in porous media: impact. Geoderma 44:95–113. http://dx.doi.org/10.1016/0016-7061(89)90021-9

45. Machat H, Carrayrou J (2017) Comparison of linear solvers for equilibrium geochemistry computations. Comput Geosci 21:131–150. https://doi.org/10.1007/s10596-016-9600-5

This article is protected by copyright. All rights reserved.

46. Wanner C, Eggenberger U, Kurz D, et al (2012) A chromate-contaminated site in southern Switzerland – Part 1: Site characterization and the use of Cr isotopes to delineate fate and transport. Appl Geochem 27:644–654. http://dx.doi.org/10.1016/j.apgeochem.2011.11.009

47. Richard FC, Bourg ACM (1991) Aqueous geochemistry of chromium: A review. Water Res 25:807–816. https://doi.org/10.1016/0043-1354(91)90160-R

Table 1: Chemical table for the test case *with only activity correction*.

	Cl^-	Ca^{2+}	Al^{3+}	Sn^{4+}	K
Cl^-	1	0	0	0	1
Ca^{2+}	0	1	0	0	1
Al^{3+}	0	0	1	0	1
Sn^{4+}	0	0	0	1	1
TOTAL (mol.l^{-1})	9.10^{-5}	10^{-5}	10^{-5}	10^{-5}	
Initial low I (mol.l^{-1})	5.10^{-7}	5.10^{-7}	5.10^{-7}	5.10^{-7}	
Initial high I (mol.l^{-1})	5.10^{-4}	5.10^{-4}	5.10^{-4}	5.10^{-4}	
$C_i = \sum_{j=1}^{Nc} a_{ij} X_j$ (mol.l^{-1})	10^{-9}	10^{-9}	10^{-9}	10^{-9}	
$T_j = \sum_{i=1}^{Nc} a_{ij} \lvert C_i \rvert$ (mol.l^{-1})	5.10^{-1}	5.10^{-1}	5.10^{-1}	5.10^{-1}	

Table 2: Initial values for the situation with a low ionic strength.

	$C_i = K \prod_{j=1}^{Nc} X_j^{a_{ij}}$ (mol.l^{-1})	$C_i = \gamma_i \lvert C_i \rvert$	$I = \frac{1}{2}\sum_{i=1}^{Nc} z_i^2 \lvert C_i \rvert$ **Fixed-point**	$I = \frac{1}{2}\sum_{i=1}^{Nc} z_i^2 \lvert C_i \rvert$ **Full Newton**
Cl^-	$5.02\ 10^{-7}$	0.997	8.41	8.41
Ca^{2+}	$5.07\ 10^{-7}$	0.987	19.00	18.58
Al^{3+}	$5.15\ 10^{-7}$	0.970	19.00	18.06
Sn^{4+}	$5.27\ 10^{-7}$	0.948	19.00	17.34

Table 3: Initial values for the situation with a high ionic strength.

	$C_i = K \prod_{j=1}^{Nc} X_j^{a_{ij}}$ (mol.l^{-1})	$C_i = \gamma_i \lvert C_i \rvert$	$I = \frac{1}{2}\sum_{i=1}^{Nc} z_i^2 \lvert C_i \rvert$ **Fixed-point**	$I = \frac{1}{2}\sum_{i=1}^{Nc} z_i^2 \lvert C_i \rvert$ **Full Newton**
Cl^-	$7.12\ 10^{-4}$	0.702	-0.82	-1.07
Ca^{2+}	$2.06\ 10^{-3}$	0.243	-0.98	-1.78
Al^{3+}	$1.20\ 10^{-2}$	$4.16\ 10^{-2}$	-0.98	-2.75
Sn^{4+}	$1.42\ 10^{-1}$	$3.51\ 10^{-3}$	-0.98	-4.12

Table 4: CPU times for 1 loop (ms)

	Outer fixed-point	Inner fixed-point	Full explicit	Full recursive

Accepted Article

This article is protected by copyright. All rights reserved.

	(Loop 1 on Figure 1)			
Only I	$1.76\ 10^{-6}$	$2.18\ 10^{-6}$	$4.70\ 10^{-6}$	$1.74\ 10^{-4}$
Phosphoric acid	$2.34\ 10^{-6}$	$2.97\ 10^{-6}$	$4.72\ 10^{-6}$	$2.91\ 10^{-4}$
Gallic acid	$2.03\ 10^{-6}$	$2.68\ 10^{-6}$	$6.30\ 10^{-6}$	$6.23\ 10^{-4}$
Iron-chromium	$3.04\ 10^{-4}$	$1.15\ 10^{-5}$	$3.59\ 10^{-5}$	$1.29\ 10^{-3}$
Morel-Morgan	$2.56\ 10^{-2}$	$2.68\ 10^{-2}$	$4.16\ 10^{-2}$	$1.27\ 10^{-1}$

Table 5: Failure ratio after 1,000 iterations

	Only-I	Phosphoric acid	Gallic acid	Iron-Chromium	Morel-Morgan
Inner fixed-point	0.00 %	0.00 %	0.00 %	1.62 %	17.91 %
Outer fixed-point	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
Full Newton	0.00 %	0.00 %	0.05 %	91.35 %	3.87 %

Accepted Article

This article is protected by copyright. All rights reserved.

Accepted Article

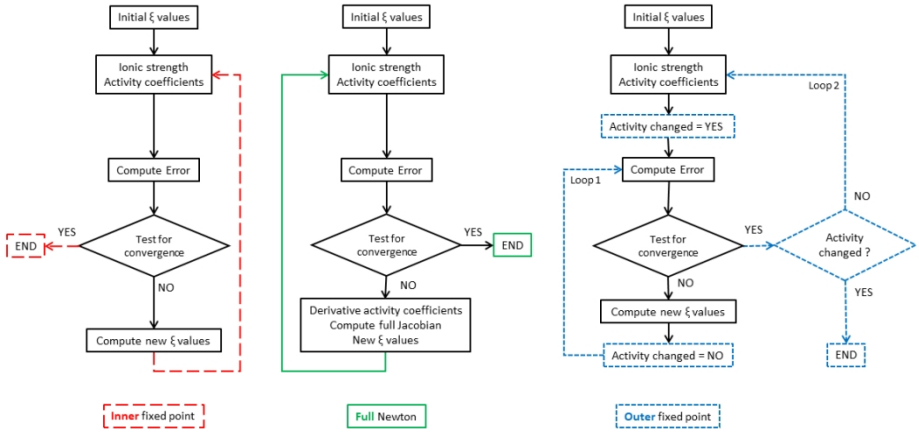

Figure 1: Inner fixed-point, full Newton and outer fixed-point activity algorithms

This article is protected by copyright. All rights reserved.

Accepted Article

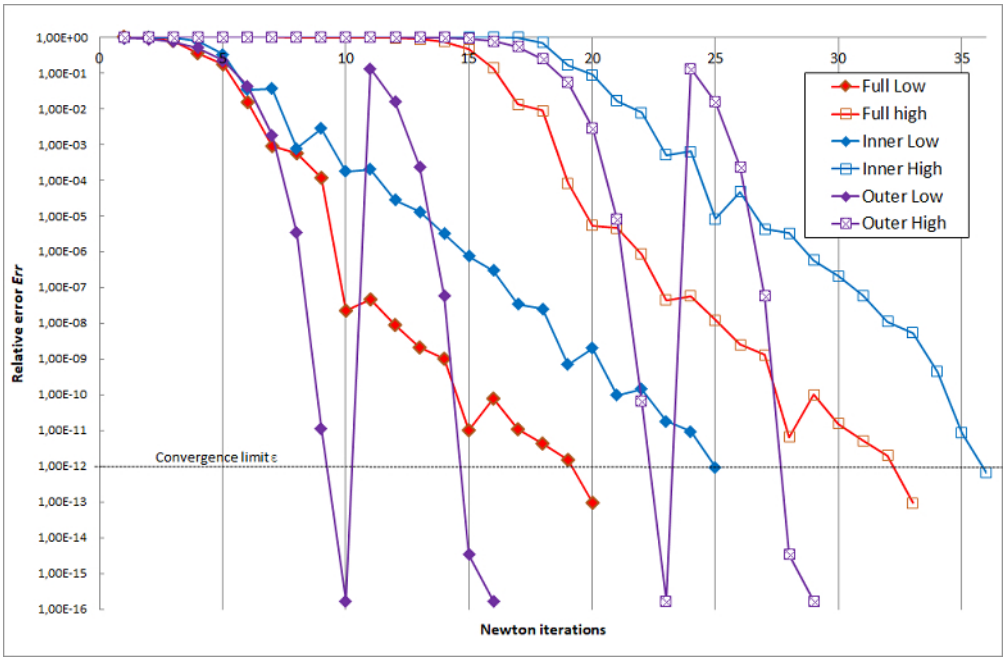

Figure 2 : Evolution of the relative error *Err* (eq 21) versus the number of Newton iterations for the test case with only activity correction

This article is protected by copyright. All rights reserved.

Accepted Article

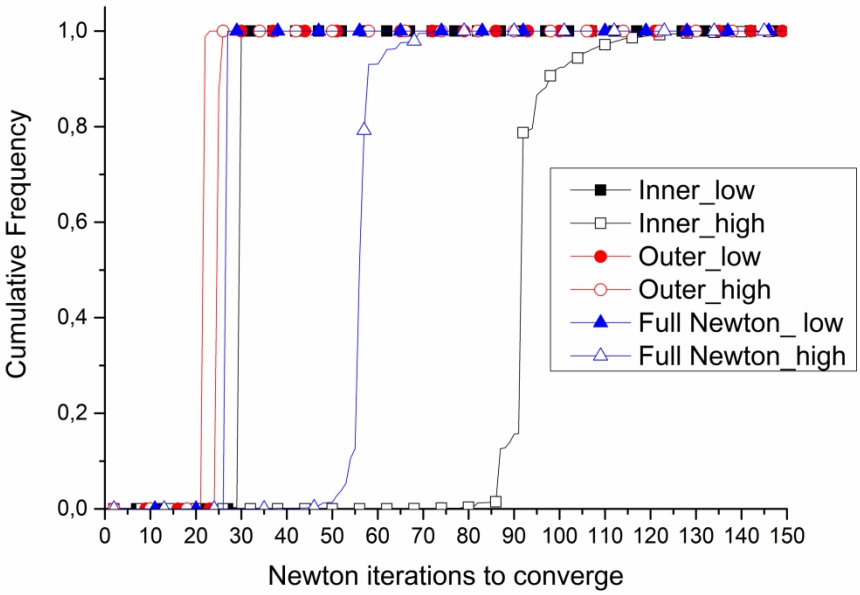

Figure 3: Frequency graph for the test case with only activity correction at low and high ionic strengths

This article is protected by copyright. All rights reserved.

Accepted Article

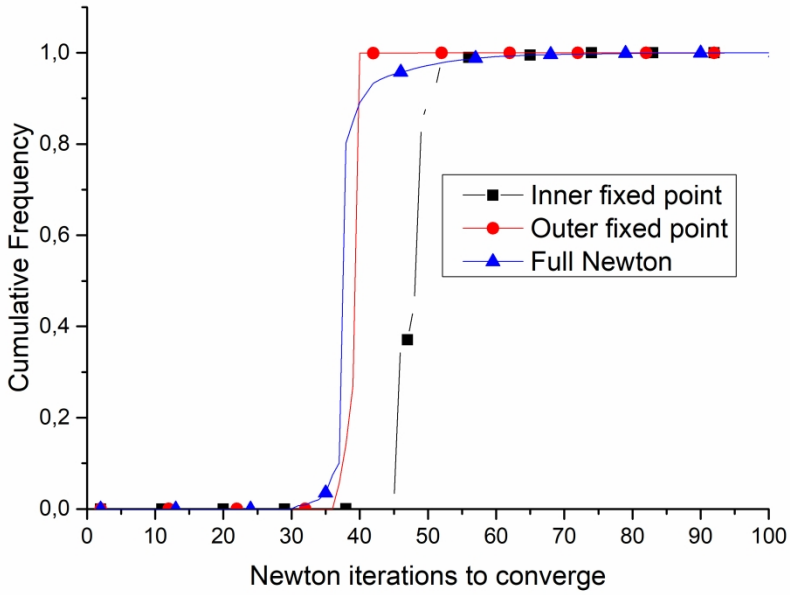

Figure 4: Frequency graph for the phosphoric acid test case

This article is protected by copyright. All rights reserved.

Accepted Article

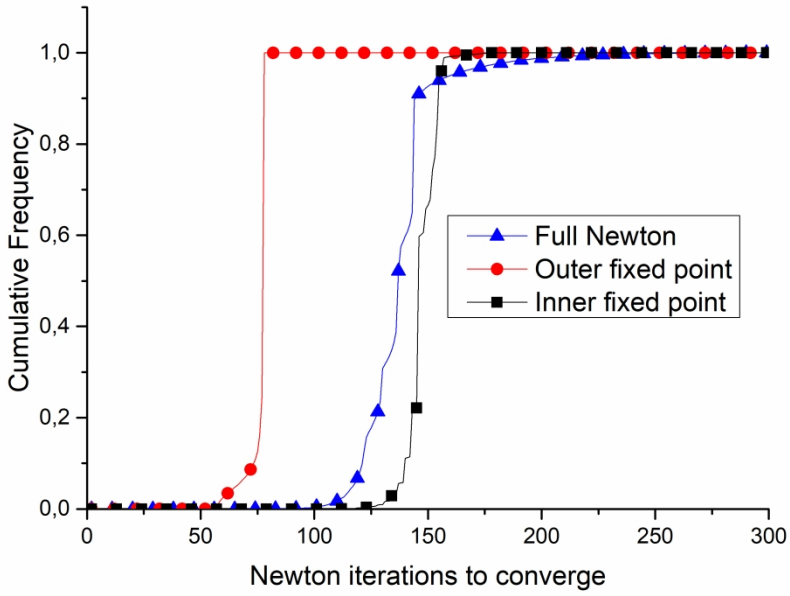

Figure 5: Frequency graph for the gallic acid test case

This article is protected by copyright. All rights reserved.

Accepted Article

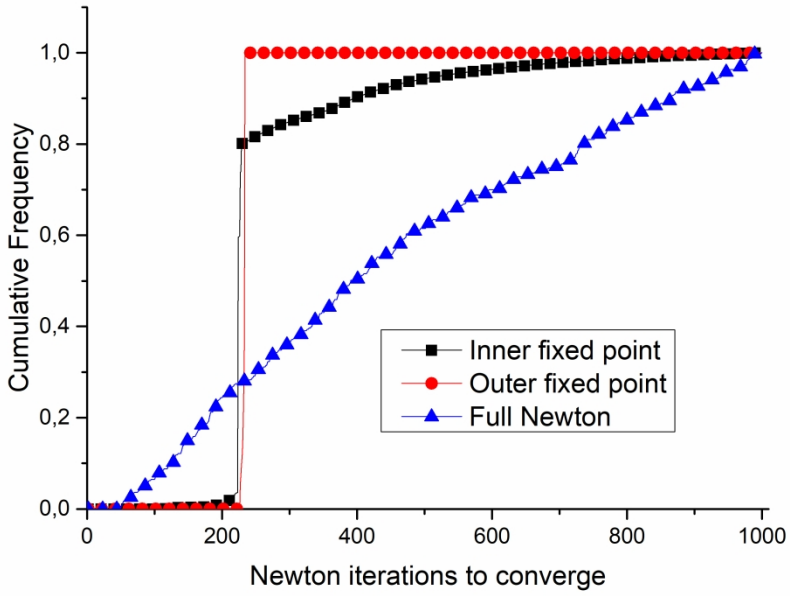

Figure 6: Frequency graph for the iron-chromium test case

This article is protected by copyright. All rights reserved.

Accepted Article

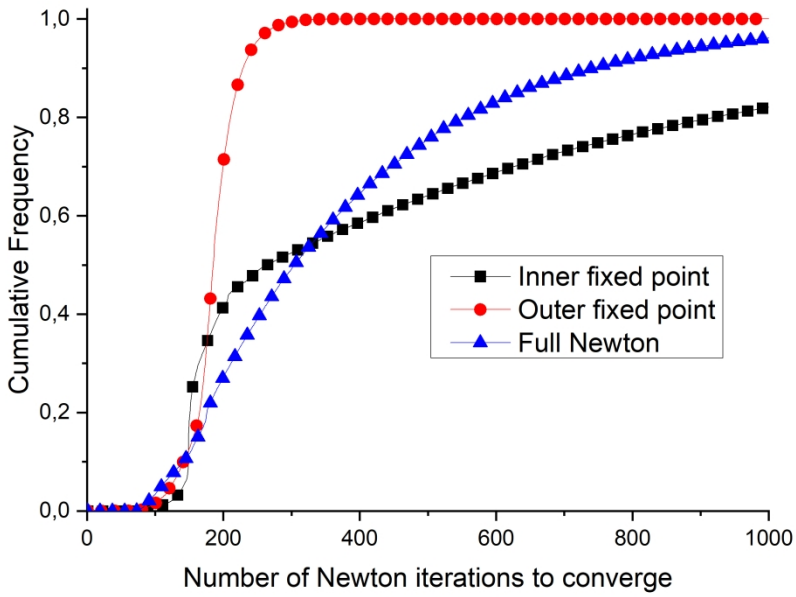

Figure 7: Frequency graph for the Morel-Morgan test case

This article is protected by copyright. All rights reserved.